PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification [6] :
G06F 9/46

A1

(11) International Publication Number: WO 99/57637

(43) International Publication Date: 11 November 1999 (11.11.99)

(21) International Application Number: PCT/GB99/01302

(22) International Filing Date: 27 April 1999 (27.04.99)

(30) Priority Data:
9809512.8 1 May 1998 (01.05.98) GB

(71) Applicant *(for all designated States except US)*: BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: LEBRE, Caroline, Anne, Michele [FR/GB]; 7 High View House, Dykes Street, Ipswich, Suffolk IP1 3NB (GB). TITMUS, Richard, John [GB/GB]; 27 King Edward Road, Ipswich, Suffolk IP4 3AN (GB).

(74) Agents: READ, Matthew, Charles et al.; Venner, Shipley & Co., 20 Little Britain, London EC1A 7DH (GB).

(81) Designated States: AU, CA, CN, JP, SG, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: DISTRIBUTED DATA PROCESSING

(57) Abstract

Data is processed in a distributed environment, using client and server objects. The server object (MS1) is mobile and can be serialised under control of a proxy (*pr1*) and moved from a first place (*P1*) to a second place (*P2*). During relocation, calls to the mobile server are frozen and redirected to the new location when the server becomes ready for use at the second place (*P2*).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

Distributed data processing

Field of the invention

This invention relates to a processing data in a distributed computing environment.

Background

Data processing may be carried out in a distributed computing environment in which client software interacts with server software, connected in a network. A server can be considered to have a resource which is to be shared with a number of clients which have an interest in it. The server waits for client initiated requests and replies to them individually with information derived from the resource requested by the client.

Conventionally, the client software is located at fixed workstations connected in the network, which interacts with servers at fixed locations. More recently, mobile agent software has been developed which allows the client software to move to a location close to a server in order to make better use of the facilities of the server. For example, if a manufacturing company has factories at two different locations, with their own local computer networks, an operator at the first location may wish to interrogate databases of servers at both locations to determine e.g. the availability of certain stock items which may be held in warehouses at the two locations. In this situation, it is convenient for the client data interrogation software to migrate from the first location to the second location in order to be close to the server at the second location, to enable the associated databases to be interrogated efficiently. The mobile client software is known as a mobile agent.

A number of different systems which provide mobile agents have been developed: MuBot by Crystaliz, Inc., Agent Tcl by Dartmouth College, Aglets by IBM, MOA by the Open Group Inc, GMAF/Magna by GMD Fokus and Odyssey by General Magic Inc.

Summary of the invention

In accordance with the invention, it has been appreciated that there are situations in which it would be advantageous to make the server mobile within a distributed computing environment.

In accordance with the invention there is provided a method of processing data in a distributed computing environment wherein a client and a server process data, the method comprising sending the server from a first place where it communicates with the client, through the distributed computing environment towards a second different place to perform data processing therefrom.

The method may include freezing incoming calls for data processing to the server at the first place whilst the server is being sent from the first place to the second place, and thereafter directing the frozen calls towards the second place to be processed by the server when it has become functional at the second place. This has the advantage of ensuring that connections are not lost to the server whilst it moves from the first place to the second place.

In another aspect, the invention includes, at the second place, receiving the server sent from the first place in order to perform data processing at the second place.

In order to transmit the server from the first place to the second place, it may be converted from an operational configuration at the first place into a configuration suitable for transmission through the distributed environment to the second place. The conversion may comprise serialisation of the server.

The invention also includes a software entity operable to provide a server for a client in a distributed computing environment characterised in that the software entity is selectively re-locatable to different places through the environment.

In another aspect, the invention includes a signal for transmission in a distributed computing environment wherein a client and a server process data, the signal comprising the server serialised for transmission between a first place where it communicates with the client, through the distributed computing environment and a second different place to perform data processing.

The transmission of the server from the first place towards the second place may be controlled by a proxy and more particularly, the invention includes a proxy for use in a distributed computing environment wherein a client and a server process data, the proxy being operable to send the server from a first place where it communicates with the client, through the distributed computing environment towards a second different place to perform data processing.

Brief description of the drawings

In order the invention may be more fully understood, an embodiment thereof will now be described way of example with reference to the accompanying drawings, in which:

Figure 1 is a schematic block diagram of a distributed computing environment which uses mobile software agents;

Figure 2 is a more detailed diagram of one of the hosts shown in Figure 1;

Figure 3 illustrates schematically the moving of a mobile server from a first place to a second place in accordance with the invention; and

Figure 4 is a schematic timing diagram of signal communication between the first place and the second place in respect of the movement of the server.

Detailed description

In the following description, the terminology adopted by the Object Management Group (OMG) for mobile agents has been adopted by way of convenient explanation. The OMG has defined a common standard for interoperability of objects between different systems under a common object management architecture that provides an object request broker known commercially as CORBA which provides an infrastructure allowing objects to

converse independently of the specific platforms and techniques used to implement the objects. In order to deal with interoperability of mobile agents, the OMG has produced a document "Mobile Agent Facility Specification" 1st September 1997, OMG TC Document orbos/97-09-20, available from the Object Management Group, 492 Old Connecticut Path, Framingham, MA 01707, USA. Members of OMG can also find the full specification at the following URL:
http://www.omg.org/library/schedule/Mobile_Agents_Facility_RFP.htm.
This will now be explained with reference to Figures 1 and 2.

For mobile software agents, which are clients, the world is made up of regions which include places between which the mobile agent can move. Referring to Figure 1, first and second host computing systems 1, 2 are interconnected by a network 3. The first and second host systems can be of any suitable form e.g. local area networks, individual computers and the like, which each operate with their own operating system *OS1*, *OS2*. In a conventional manner, the individual hosts 1, 2, may include one or more computers or processors, each of which include a processor, volatile working memory and non-volatile data storage. Each host is provided with a communications interface *CI1*, *CI2* to allow communication between them via the network 3. The network 3 can be of any suitable form, for example a wide area network, a local area network, intranet or Internet.

Considering the host 1, its operating system *OS1* provides an environment in which software can operate. The client software is configured as mobile software agent *MA1*. Similarly, the host 2 has an operating system *OS2* and a mobile agent *MA2*. A further mobile agent *MAn* is shown in host 1. Each mobile agent *MA* is operative at a place *P*. Thus, considering the host 1, mobile agent *MA1* is operative at place *P1* and mobile agent *MAn* is operative at place *Pn*. Mobile agent *MA2* is at place *P2* in host 2. The mobile agents can move from place to place. It will be understood that in host 1, the places *P* may be individual computers connected in a network that comprises host 1 or any other

suitable hardware configuration, which will not be described further herein. The same is true for host 2. The OMG mobile agent specification is designed to provide interoperability between different operating systems in order to allow transport of mobile client agents from one host to another. It is assumed in the configuration of Figure 1 that different operating systems *OS1, OS2* are in use although this is not an essential feature of the invention. It will be appreciated that the OMG specification makes use of CORBA to allow interoperability between different hardware and software configurations. The agents operating within the operating system *OS1* define an agent system *AS1* in host 1. A similar agent system *AS2* operates in host 2 shown in Figure 1.

The software process is arranged in a client-server configuration as will now be explained with reference to Figure 2. Conveniently, but not necessarily, the software may be object oriented such that the mobile client agents and the servers can each be considered as objects. As shown in Figure 2, server software *MS 1* is shown at place *P1* which can service calls from the mobile clients described with reference to Figure 1. For example, the mobile agent *MA1* is a client at place *P1* and can make data calls on the server *MS 1* over path 4 to perform data processing. The client and server do not however need to be located at the same place *P*. Thus, in the example of Figure 2, the server *MS1* can service data calls from mobile client agent *MA 2* at place *P2* over communication path 5. It will be understood that there may be more than one server *MS* in the distributed computing environment.

In accordance with the invention, the server *MS 1* is mobile within the distributed computing environment. In order to manage the mobility of the mobile server *MS 1* , it is given a software proxy *pr 1* which is different in each place *P*. The proxy *pr 1* is advertised to CORBA with the mobile server interface, instead of the mobile server itself. All processing calls for the server go to the proxy first and are then redirected by it to the server. Therefore, the proxy *pr 1* knows at all times how many clients are connected to the server *MS 1* and how many calls are in progress.

Referring to Figure 3, there are situations where it would be convenient to move the server agent *MS1* from place *P1* to place *P2* via the communication interface *CI1*, network 3 and interface *CI2*. For example, the server *MS 1* could then function with enhanced operability with client *MA 2* residing in the agent system *AS2* in host 2. The transfer of the server *MS 1* from place *P1* to place *P2* will now be described in detail with reference to Figure 4.

Initially, when the mobile server *MS 1* decides or is told to move from place *P1*, at step S.0, it tells its proxy *pr1* the place to which it is to be moved. In this case, the mobile server *MS1* is to be moved to place *P2*. Alternatively, the proxy *pr1* may be told by some external third party to move the mobile server. The moving process then starts.

At step S.1, the proxy *pr1* freezes all incoming calls for data processing to the mobile server *MS 1*.

At step S.2, the proxy *pr1* waits until all current data processing handled by mobile server *MS 1* has finished.

Then, at step S.3, the proxy *pr1* tells mobile server *MS 1* that it is about to be moved and that it must perform any task needed to be completed before leaving place *P1*.

Then, at step S.4, the proxy causes the mobile server *MS 1* to be serialised, namely to convert it from its operational state into a condition suitable for transmission through the network 3 (Figure 1).

Then, at step S.5, the serialised mobile server is sent to the place *P2* of host 2 via communications interface CI1, network 3 and communications interface CI2.

At step S.6, a new proxy *pr1'* is produced in place *P2* for the mobile server *MS1*

when located in place *P2*.

At step S.7, the mobile server *MS1* is de-serialised at place *P2* and thereby recreated in an operational condition.

At step S.8, the newly created proxy *pr1'* sends back locational reference data for mobile server *MS1*, so as to indicate to the proxy *pr1*, the new CORBA reference of mobile server *MS1*.

Then, at step S.9, the calls frozen at step S.1 are forwarded to the mobile server *MS1* through the network 3, by proxy *pr1*, from place *P1* to place *P2*.

The procedure described with reference to Figure 4 has the advantage that communication with the mobile server *MS 1* is not lost during the transfer process. The steps ensure that any data processing carried out at place *P1* is completed before the transfer occurs and, whilst the transfer is taking place, incoming calls are frozen and then transferred to the new place.

Clients can find the moved mobile server *MS 1* by making an appropriate request, as for any other CORBA object, and will receive the reference of its proxy. The proxy that is advertised for the mobile agent can either be the first one *pr1*, in which case calls will be directed from *pr1* to *pr1'*, or *pr1'* itself.

At the completion of the moving process for the mobile server, the proxy *pr 1* is no longer needed and is cancelled.

It will be understood that client agents such as agent *MA*2 shown at place *P1* in Figure 3 can be mobile in a conventional manner, in accordance with the OMG specification for mobile agents. Thus, client agent *MA2* can be moved in a conventional manner by serialising the agent, transmitting it through the network 3 to a different place and de-serialising the agent at the new place. Thus, it is possible according to the invention to move an entire client - server

combination from one place to another or to different places.

It will be understood that the mobile server MS1 when at a particular place, will reside in the working memory of a particular computer within the host, and may if required be stored in the non-volatile memory of the computer associated with the place *P*, to provide a record thereof if the network or a part thereof is shut down. The mobile server may also be provided on a storage medium such as an optical or magnetic disc, so that it can be loaded into a computer at a particular place *P*, and then commence its mobile activities in the network.

Whilst the previously described clients and servers may conveniently be configured as software objects in an object oriented environment, this is not essential and they can be configured as batches of conventional code. Also, whilst the invention has been described in relation to CORBA object management architecture, other management architectures could be used such as OLE by Microsoft, suitably configured to handle mobile objects

Movement of the server in accordance with the invention renders the computing process much more flexible. For example in an Internet application, if a large number of clients in the United Kingdom are calling a server which is located at a place in the USA, a large number of transatlantic calls would need to be set up, leading to inefficiencies. In accordance with the invention, the server object can migrate from a place in the USA to a place in the United Kingdom, speeding up execution of the individual client/server processes.

Claims

1. A method of processing data in a distributed computing environment wherein a client and a server process data, the method comprising sending the server from a first place where it communicates with the client, through the distributed computing environment towards a second different place to perform data processing therefrom.

2. A method according to claim 1 including freezing incoming calls for data processing to the server at the first place whilst it is being sent from the first place to the second place, and thereafter directing the frozen calls towards the second place to be processed by the server when it has become functional at the second place.

3. A method according to claim 2 including waiting for the server to complete its current processing tasks before sending it to the second place.

4. A method according to any preceding claim including converting the server from an operational configuration at the first place into a configuration suitable for transmission through the distributed environment to the second place.

5. A method according to claim 4 wherein the conversion comprises serialisation of the server.

6. A method according to any preceding claim including creating a proxy for the server at the first place, which controls the sending of the server towards the second place.

7. A method according to any preceding claim including sending the client towards a different place in the distributed computing environment.

8. A method of processing data in a distributed computing environment wherein a client and a server process data, the method comprising receiving the server sent from a first place where it communicated with the client, through the distributed computing environment, at a second different place, to perform data processing at the second place.

9. A method according to claim 8 wherein the server is received at the second place in a form suitable for transmission through the distributed environment, and including converting the received server at the second place into a form suitable for processing data at the second place.

10. A method according to claim 9 wherein the converting includes deserialising the server.

11. A method according to claim 8, 9 or 10 including producing a proxy for the received server, at the second place.

12. A method according to any one of claims 8 to 11 including receiving at the second place, data processing calls for the server directed thereto from the first place after the server has become operational at the second place.

13. A software entity operable to provide a server for a client in a distributed computing environment **characterised in that** the software entity is selectively re-locatable to different places through the environment.

14. An entity according to claim 13, operable to function as the server at a first place in the environment and then to re-locate and function as the server at a second place in the environment.

15. An entity according to claim 13 or 14, operable such that data calls thereto from a client are frozen during the re-location.

16. An entity according to any one of claims 13 to 15 operable to provide a proxy functional to send the server through the environment to achieve the re-location.

17. An entity according to claim 16 wherein the proxy is functional to wait for the server to complete its current processing tasks before commencing the re-location.

18. An entity according to claim 16 or 17 wherein the proxy is operable to serialise the server from its functional configuration into a configuration suitable for transmission through the distributed environment so as to achieve the re-location.

19. A software entity according to any one of claims 13 to 18, stored on a storage medium.

20. A signal for transmission in a distributed computing environment wherein a client and a server process data, the signal comprising the server serialised for transmission between a first place where it communicates with the client, through the distributed computing environment and a second different place to perform data processing.

21. A proxy for use in a distributed computing environment wherein a client and a server process data, the proxy being operable to send the server from a first place where it communicates with the client, through the distributed computing environment towards a second different place to perform data processing.

22. A proxy according to claim 21 operable to freeze incoming calls for data processing to the agent at the first place whilst it is being sent from the first place to the second place, and thereafter to direct the frozen calls towards the second place to be processed by the server when it has become functional at the second

place.

23. A proxy according to claim 21 or 22 operable to wait for the server to complete its current processing tasks before sending it to the second place.

24. A proxy according to claim 21, 22 or 23 operable to serialise the server from an operational configuration at the first place into a configuration suitable for transmission through the distributed environment to the second place.

25. A host provided with client and server objects for processing data in an object oriented distributed processing environment **characterised in that** the server object is selectively re-locatable to different places in the environment.

26. A host according to claim 25 wherein the mobile server object is operable such that data calls thereto are frozen during the relocation.

27. A host according to claim 25 wherein the server is provided with a proxy compatible with CORBA or OLE architecture.

28. A server object for processing data in an object oriented distributed processing environment **characterised in that** the server object is re-locatable for operation at different places and is provided in use with a proxy which freezes data calls thereto during the relocation and subsequently forwards them to the moved server object.

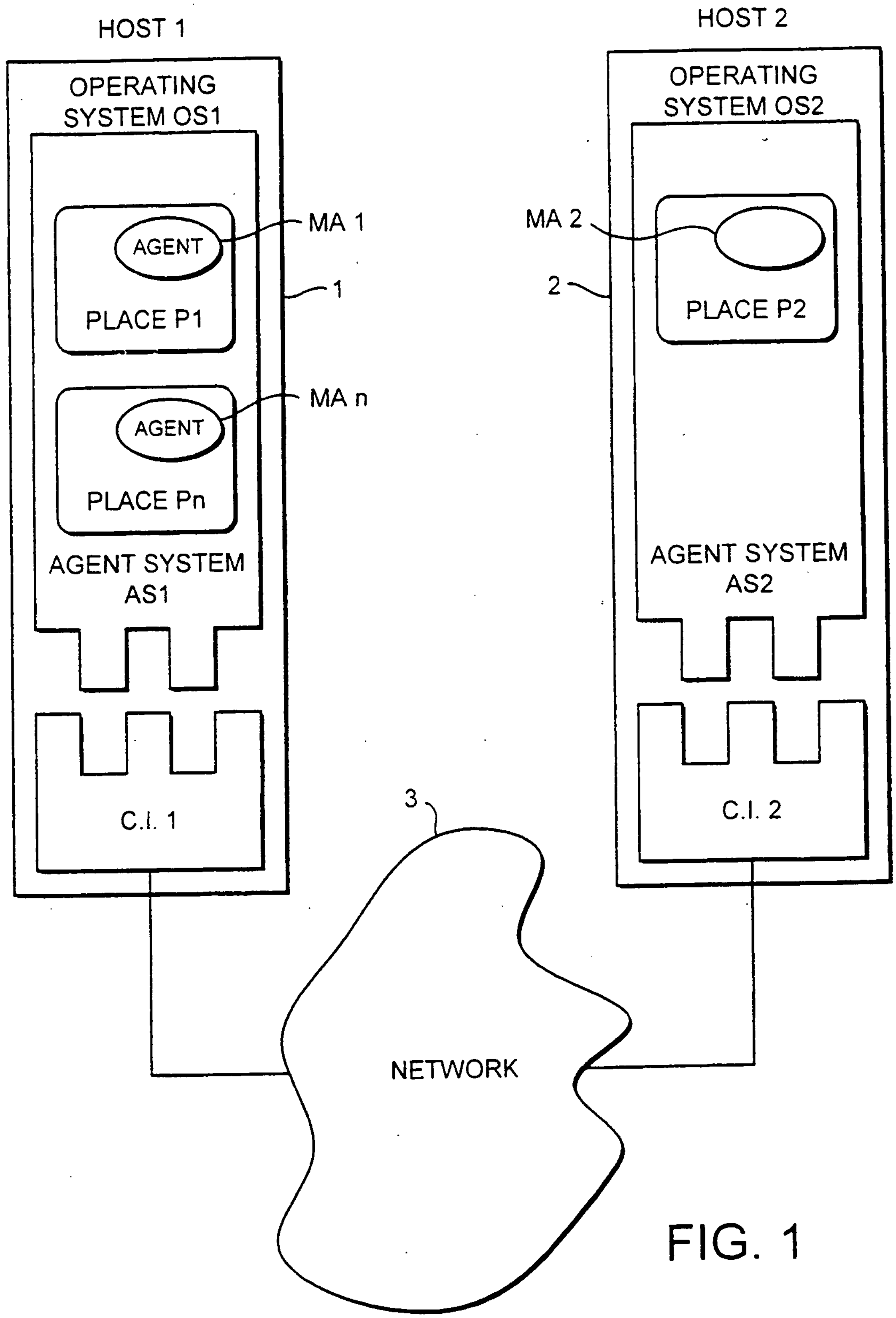

HOST 1
OPERATING SYSTEM OS1
AGENT
MA 1
PLACE P1
1
AGENT
MA n
PLACE Pn
AGENT SYSTEM AS1
C.I. 1
HOST 2
OPERATING SYSTEM OS2
MA 2
2
PLACE P2
AGENT SYSTEM AS2
C.I. 2
3
NETWORK

FIG. 1

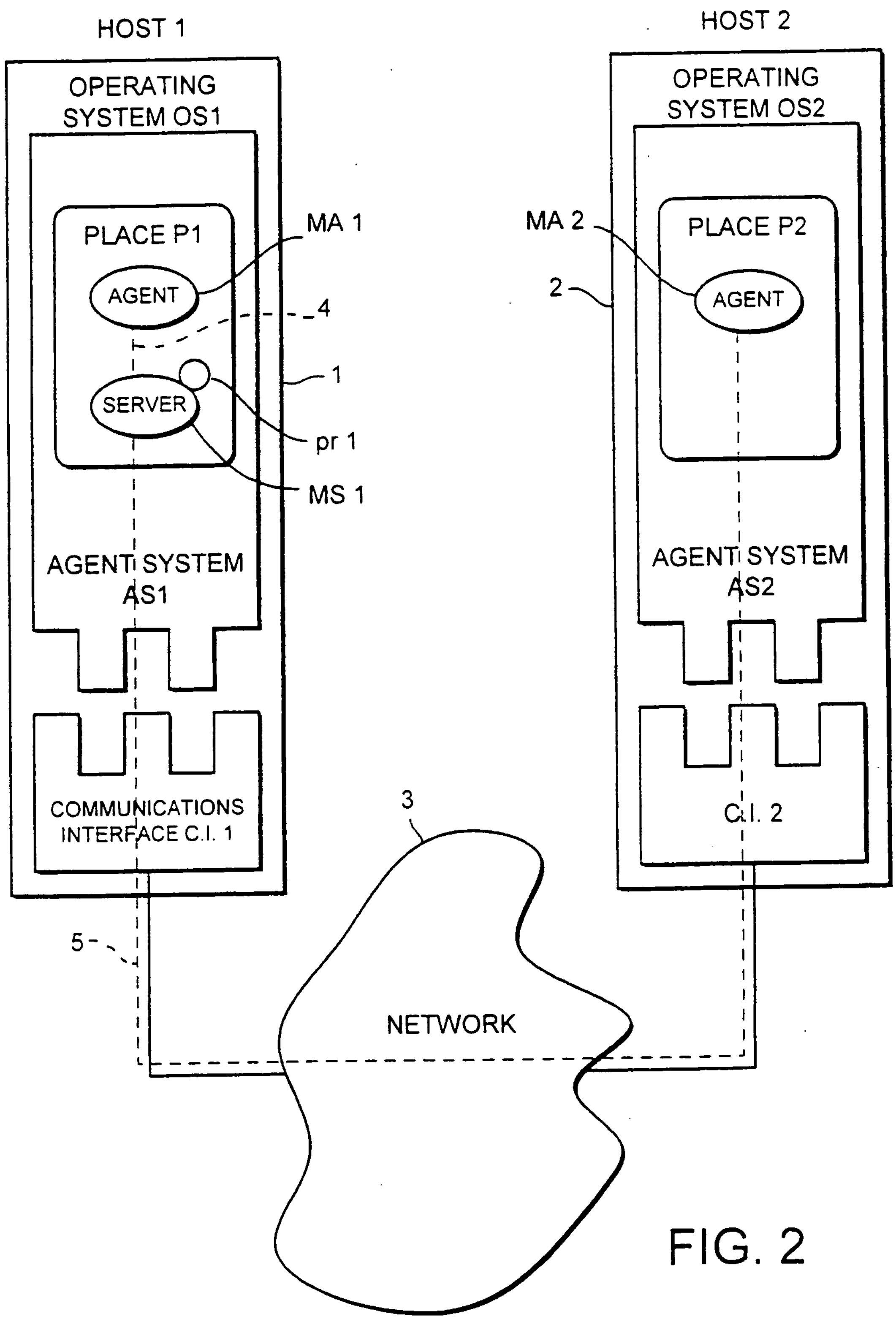

HOST 1
OPERATING SYSTEM OS1
PLACE P1
AGENT
SERVER
MA 1
4
1
pr 1
MS 1
AGENT SYSTEM AS1
COMMUNICATIONS INTERFACE C.I. 1
5
HOST 2
OPERATING SYSTEM OS2
MA 2
2
PLACE P2
AGENT
AGENT SYSTEM AS2
C.I. 2
3
NETWORK

FIG. 2

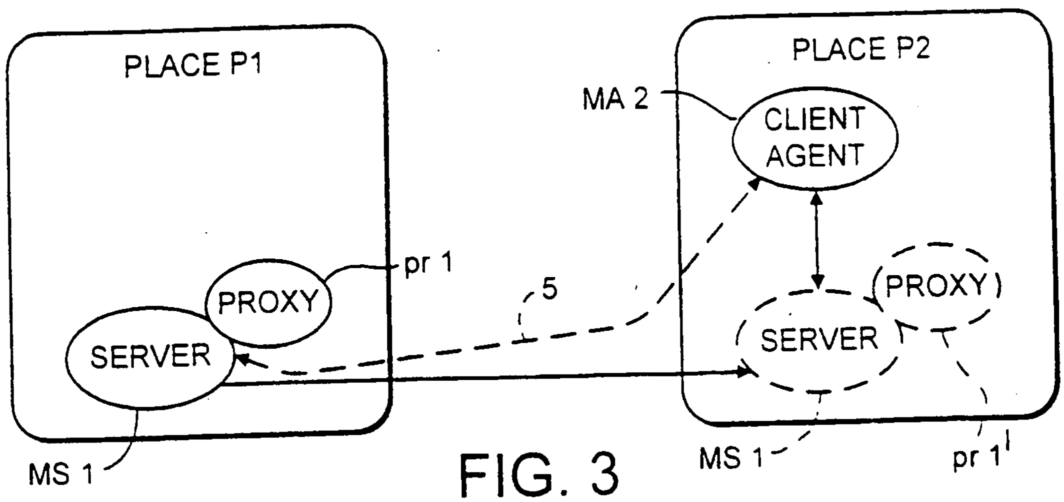

PLACE P1
PROXY
pr 1
SERVER
MS 1
5
PLACE P2
MA 2
CLIENT AGENT
PROXY
SERVER
MS 1
pr 1'

FIG. 3

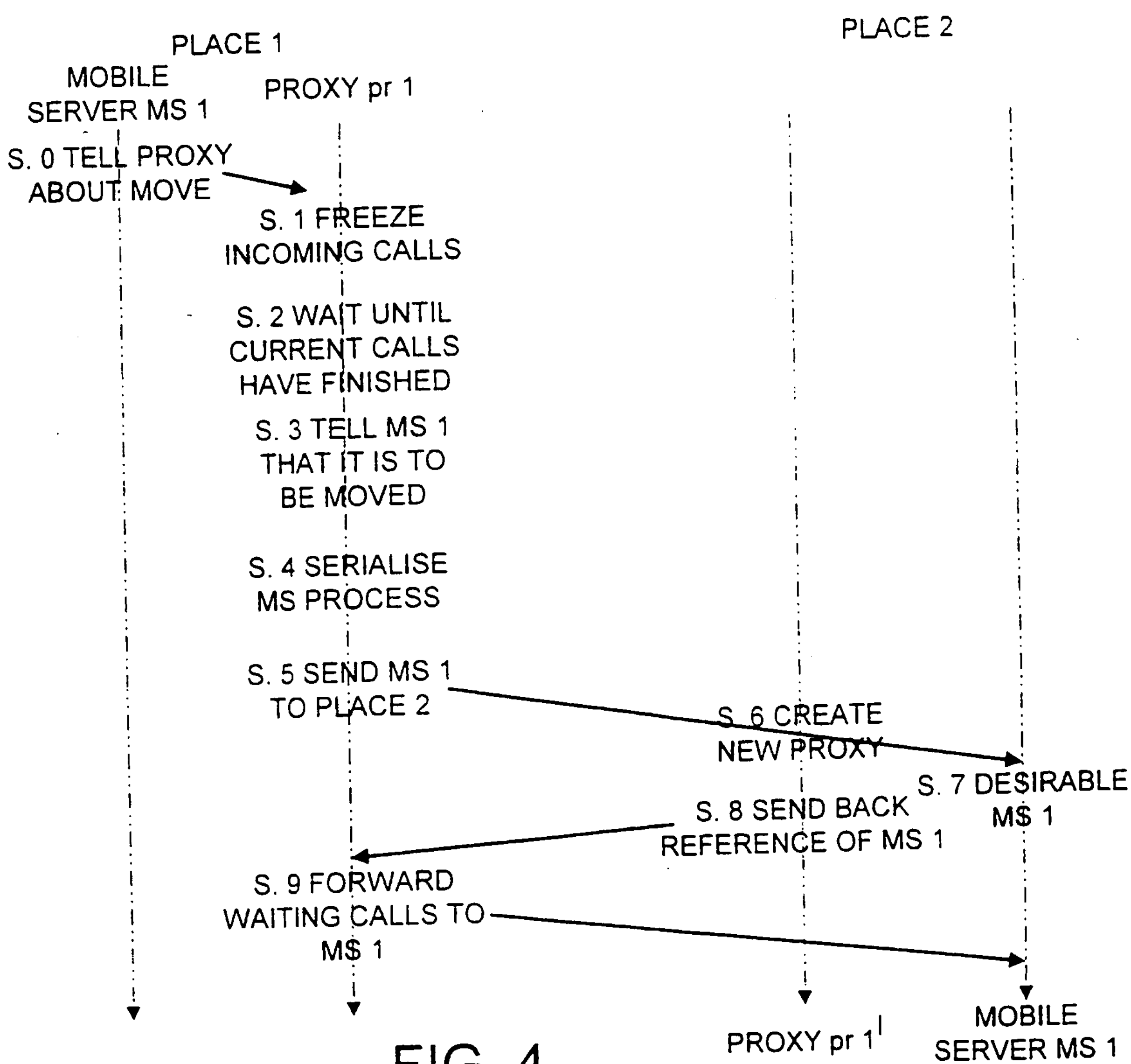

PLACE 1
MOBILE SERVER MS 1
PROXY pr 1
PLACE 2
S. 0 TELL PROXY ABOUT MOVE
S. 1 FREEZE INCOMING CALLS
S. 2 WAIT UNTIL CURRENT CALLS HAVE FINISHED
S. 3 TELL MS 1 THAT IT IS TO BE MOVED
S. 4 SERIALISE MS PROCESS
S. 5 SEND MS 1 TO PLACE 2
S. 6 CREATE NEW PROXY
S. 7 DESIRABLE MS 1
S. 8 SEND BACK REFERENCE OF MS 1
S. 9 FORWARD WAITING CALLS TO MS 1
PROXY pr 1'
MOBILE SERVER MS 1

FIG. 4

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 99/01302

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F9/46

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	BERGHOFF J ET AL: "Agent-based configuration management of distributed applications" PROCEEDINGS. THIRD INTERNATIONAL CONFERENCE ON CONFIGURABLE DISTRIBUTED SYSTEMS (CAT. NO.96TB100046), PROCEEDINGS OF INTERNATIONAL CONFERENCE ON CONFIGURABLE DISTRIBUTED SYSTEMS, ANNAPOLIS, MD, USA, 6-8 MAY 1996, pages 52-59, XP002086942 ISBN 0-8186-7395-8, 1996, Los Alamitos, CA, USA, IEEE Comput. Soc. Press, USA the whole document ---	1-28
A	WO 97 47091 A (ADOBE SYSTEMS INC) 11 December 1997 (1997-12-11) the whole document ---	1,8,13, 14,20, 25,28

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
22 July 1999	29/07/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Fonderson, A

1

Form PCT/ISA/210 (second sheet) (July 1992)

International Application No
PCT/GB 99/01302

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	RAVI T M ET AL: "A basic protocol for routing messages to migrating processes" PROCEEDINGS OF THE 1988 INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, UNIVERSITY PARK, PA, USA, 15-19 AUG. 1988, pages 188-197 vol.2, XP002090982 ISBN 0-271-00654-4, 1988, University Park, PA, USA, Pennsylvania State Univ, USA the whole document -----	1,2, 13-15, 21,22, 25,26,28

1

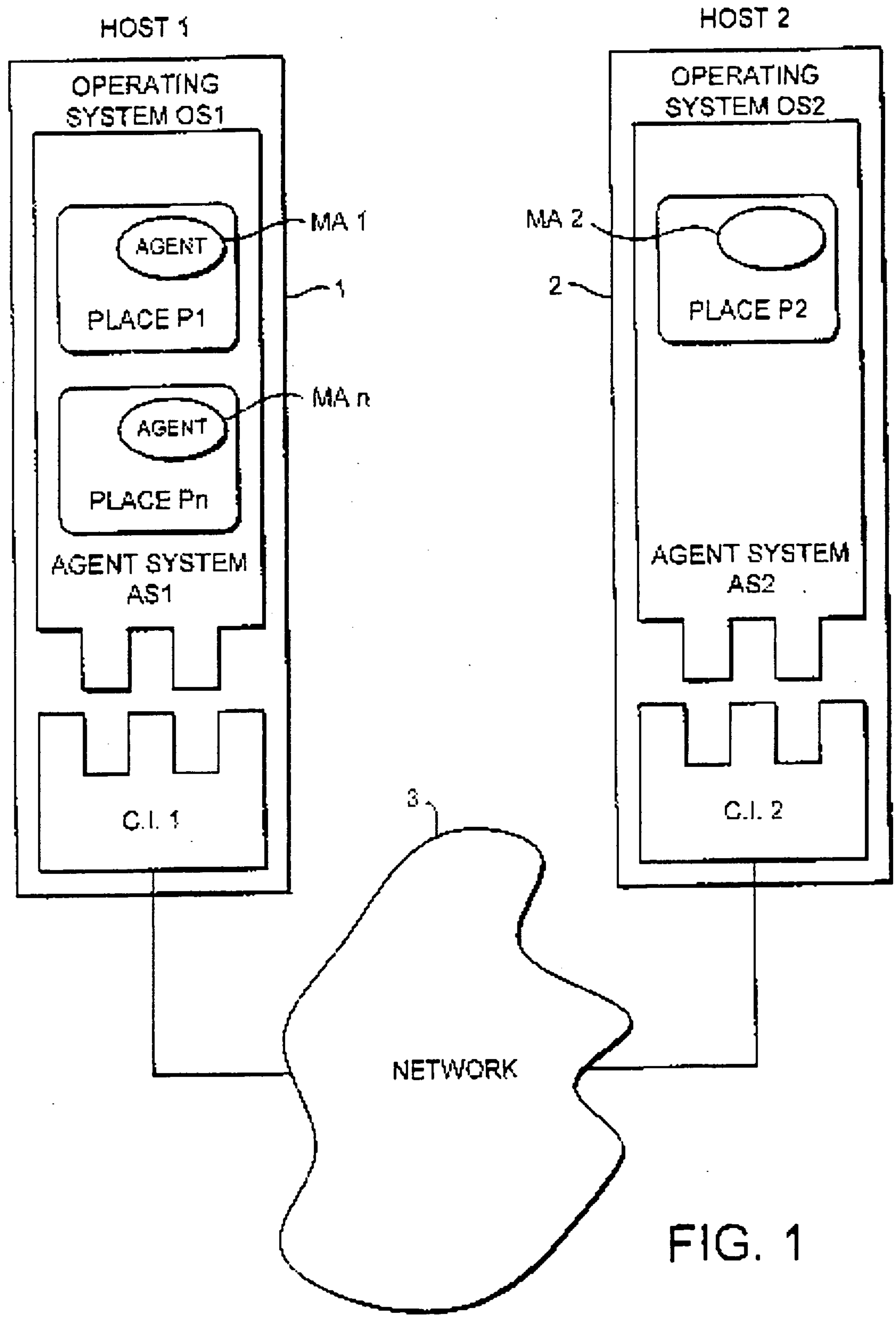
INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/GB 99/01302

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9747091 A	11-12-1997	US 5838909 A	17-11-1998
		AU 3145897 A	05-01-1998

Form PCT/ISA/210 (patent family annex) (July 1992)



HOST 1
OPERATING SYSTEM OS1
AGENT
MA 1
PLACE P1
1
AGENT
MA n
PLACE Pn
AGENT SYSTEM AS1
C.I. 1
HOST 2
OPERATING SYSTEM OS2
MA 2
2
PLACE P2
AGENT SYSTEM AS2
C.I. 2
3
NETWORK

FIG. 1

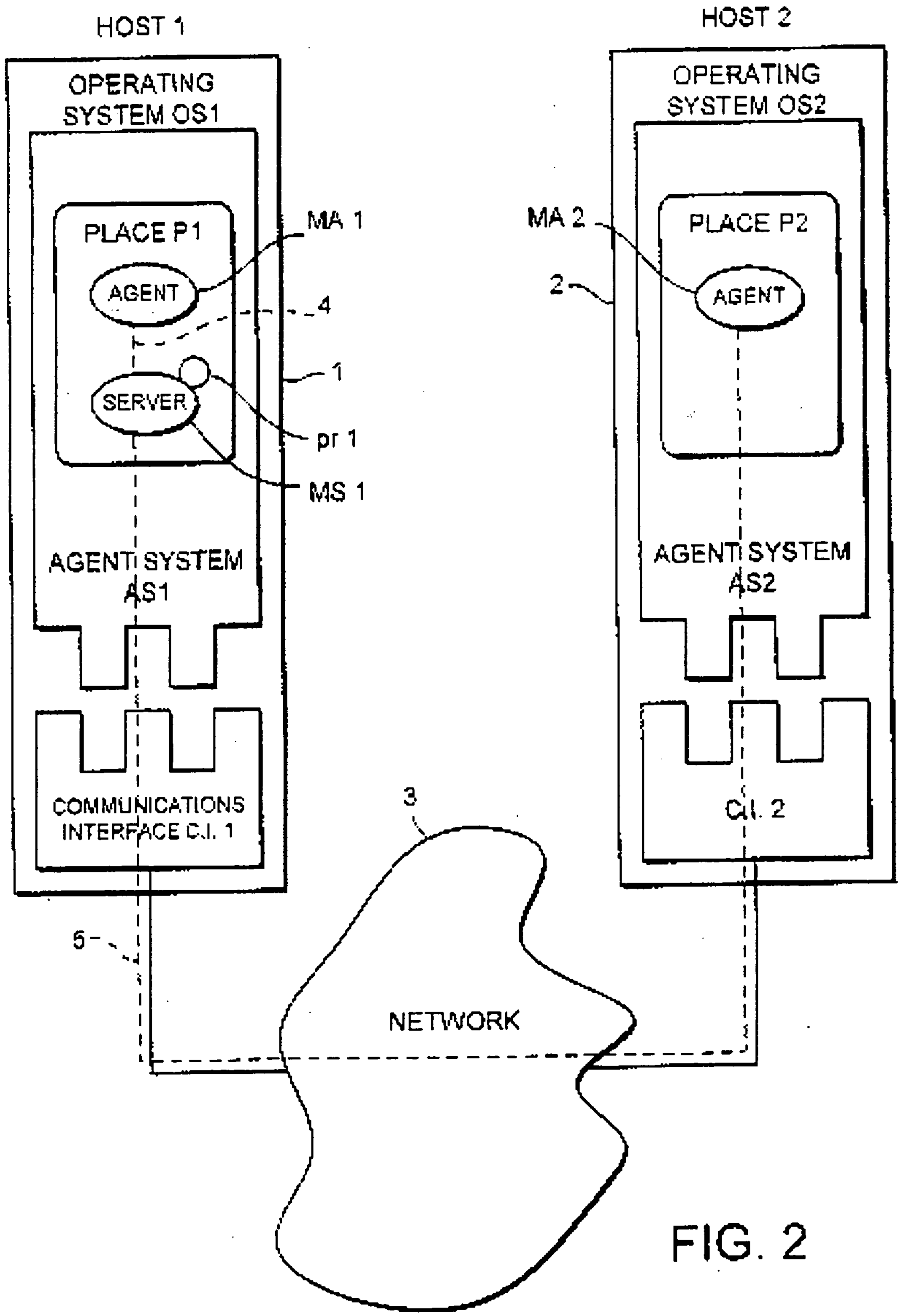

HOST 1
OPERATING SYSTEM OS1
PLACE P1
AGENT
SERVER
MA 1
4
1
pr 1
MS 1
AGENT SYSTEM AS1
COMMUNICATIONS INTERFACE C.I. 1
5
HOST 2
OPERATING SYSTEM OS2
PLACE P2
MA 2
2
AGENT
AGENT SYSTEM AS2
C.I. 2
3
NETWORK

FIG. 2

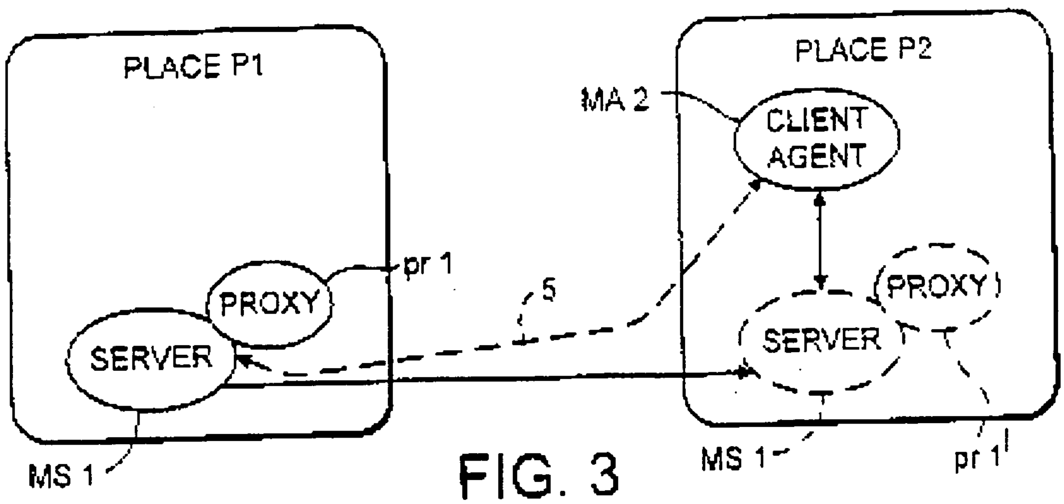

PLACE P1
PLACE P2
MA 2
CLIENT AGENT
pr 1
PROXY
5
PROXY
SERVER
SERVER
MS 1
MS 1
pr 1'

FIG. 3

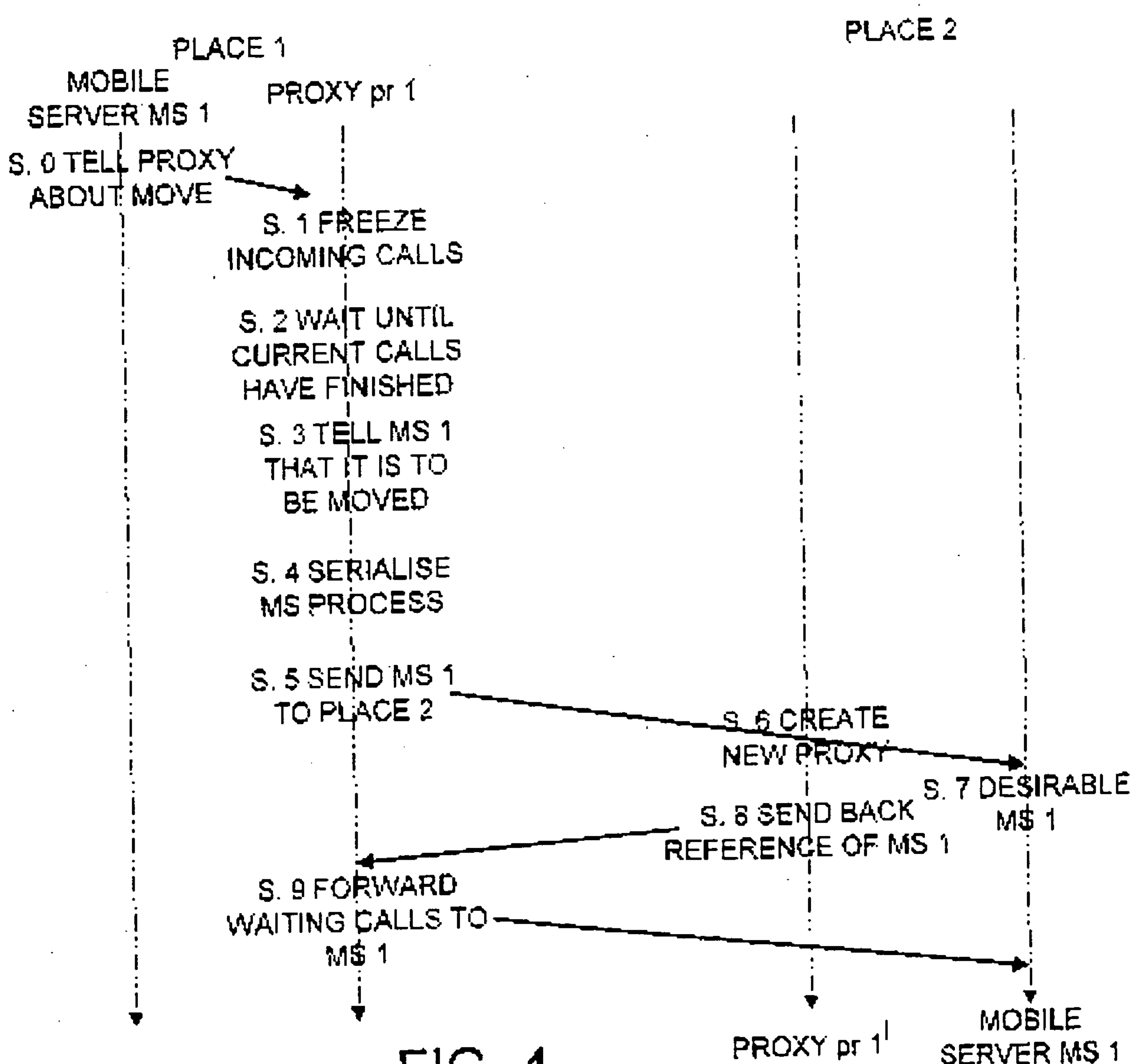

PLACE 1
PLACE 2
MOBILE SERVER MS 1
PROXY pr 1
S. 0 TELL PROXY ABOUT MOVE
S. 1 FREEZE INCOMING CALLS
S. 2 WAIT UNTIL CURRENT CALLS HAVE FINISHED
S. 3 TELL MS 1 THAT IT IS TO BE MOVED
S. 4 SERIALISE MS PROCESS
S. 5 SEND MS 1 TO PLACE 2
S. 6 CREATE NEW PROXY
S. 7 DESIRABLE MS 1
S. 8 SEND BACK REFERENCE OF MS 1
S. 9 FORWARD WAITING CALLS TO MS 1
PROXY pr 1'
MOBILE SERVER MS 1

FIG. 4